Intelligent Systems For Aerospace Engineering--An Overview

K. KrishnaKumar
NeuroEngineering Laboratory
NASA Ames Research Center
kkumar@mail.arc.nasa.gov

1 Abstract

Intelligent systems are nature-inspired, mathematically sound, computationally intensive problem solving tools and methodologies that have become extremely important for advancing the current trends in information technology. Artificially intelligent systems currently utilize computers to emulate various faculties of human intelligence and biological metaphors. They use a combination of symbolic and sub-symbolic systems capable of evolving human cognitive skills and intelligence, not just systems capable of doing things humans do not do well. Intelligent systems are ideally suited for tasks such as search and optimization, pattern recognition and matching, planning, uncertainty management, control, and adaptation. In this paper, the intelligent system technologies and their application potential are highlighted via several examples.

2 Defining Intelligent Systems

Science has evolved with our efforts towards understanding and mimicking nature, through inventions and discoveries, hypotheses and proofs, success and failures. The evolution of computers marks the era of our success in building systems that can perform actions of a repetitive kind, those which are difficult or time consuming if done by humans. This has helped enhance our efforts towards studying and understanding the intelligence of biological systems and applying this knowledge towards building artificially intelligent systems.

We attribute intelligence to various faculties. Intelligence is often identified in terms of competence, expertise, talent, schooling, IQ, and social interaction. From the perspective of computation, the intelligence of a system is characterized by its flexibility, adaptability, memory, learning, temporal dynamics, reasoning, and the ability to manage uncertain and imprecise information. Capabilities such as information gathering, understanding, making inferences, and applying it to understand and solve new problems efficiently are observed to be critical features of such systems. Intelligence has been defined in several ways:

- the ability to learn or understand from experience
- ability to acquire and retain knowledge
- mental ability
- the ability to respond quickly and successfully to a new situation
- use of the faculty of reason in solving problems, directing conduct, etc. effectively

Although Artificial Intelligence (AI) is the corner stone on which Intelligent System (IS) technologies are built, there are distinguishing differences between AI and IS. These differences are highlighted next.

AI has evolved with two distinct dimensions (Russell and Norvig, 1995), one is "humanistic AI" and the other is "rationalistic AI". Humanistic AI examines machines that think and act like-humans, whereas rationalistic AI examines machines that can be built on the understanding of intelligent human behavior. Some definitions of AI related to these ideas are presented below.

Humanistic AI
"The art of creating machines that perform functions that require intelligence when performed by people" (Kurzweil, 1990)
"The study of how to make computers do things at which, at the moment, people are better" (Rich and Knight, 1991)

Rationalistic AI
"A field of study that seeks to explain and emulate intelligent behavior in terms of computational processes" (Schalkoff, 1990)
"The branch of computer science that is concerned with the automation of intelligent behavior" (Luger and Stubblefield, 1993)

Intelligent systems as envisioned today are mostly modeled after rationalistic AI. They examine intelligent behavior using the models of human systems that enable intelligent behavior. In addition, IS has distinguished itself from AI by including intelligent behavior as seen in nature as a whole. This intelligent behavior includes biological genetics (distributed information), evolution (survivability), chaos (structured randomness), and natural adaptation (sufficability). Also, innovations in IS are driven by the need to solve complex problems with improving efficiencies. This improvement could be over a period of computing time, real-time, or over a set of problems encountered. The idea is not to converge to an optimal solution but to find solutions that are better than their predecessors. The underlying assumption here is that the problem domain is either too enormous with few good solutions or is non stationary.

So how can one define intelligent systems? This is indeed a difficult question and is subject to a great deal of debate. From the author's view point, an intelligent system is one that emulates some aspects of intelligence exhibited by nature. These include:

1. Learning
2. Adaptability
3. Robustness across problem domains
4. Improving efficiency (over time and/or space)
5. Information compression (data to knowledge)
6. Extrapolated reasoning

The debate of what is an intelligent system will be around as long as the definition of intelligence itself eludes us. From the practical view point, IS technologies have made it easier to achieve some level of complexity in aerospace applications just as much computers have enabled certain level of complexity

3 Role of Intelligent Systems in Aerospace Engineering

Intelligent System (IS) applications have gained popularity among aerospace professionals in the last decade due to the ease with which several of the IS tools can be implemented. In addition to

this ease of implementation, IS has been shown to solve difficult problems more efficiently. Another advantage that IS practitioners have seen is complex ideas can be implemented and tested with rapid development cycles. All of these rely heavily on the ubiquitous nature of computing machines with the power of 20th century supercomputers. The applications have gained popularity among the technical and user communities for both intellectual curiosity and for practical reasons. Some of the novel ideas using IS include spacecraft autonomy, aircraft control, modeling, airfoil design, satellite operations, missile design, and vehicle health management. In essence one can say that the IS technologies help achieve efficiency, robustness in addition to providing human-like capabilities such as pattern recognition, learning, long-term optimization, planning, and self-improvement.

The role of intelligent systems in aerospace engineering is two-fold: (1) function as intelligent assistants to augment human expertise; and (2) act as a substitute for human expertise in endeavors that save cost, time, and life. For example, intelligent systems assist humans in solving difficult optimization problem by their shear ability to robustly search through myriad of choices. In contrast, intelligent systems are used on autonomous rovers to both save cost and human lives.

In the next several sections, we present some areas of applications using intelligent systems and highlight the benefits using examples related to aerospace engineering.

3.1 Intelligent Systems for Modeling

Modeling could be thought of as a representation of available information. Intelligent systems provide two very important features for modeling: generalization and robustness. Generalization implies that the model could be used not only to represent just the data gathered but the knowledge the data represents. Robustness can be defined as the system's ability to perform within certain bounds of its nominal (without uncertainty) performance in the presence of bounded uncertainty. Several techniques such as neural networks, fuzzy logic, expert systems, etc have been routinely used by aerospace engineers for modeling. Knowledge representation in general contains syntax and semantics. Syntax is the constitution of sentences and Semantics is the interpretation of sentences. Examples of knowledge representation include

- **Mathematical Equations (Ex: ARMA Models**: Autoregressive moving average models are created using the least-square technique to represent linear relationship between inputs and outputs.)
- **Rule-based systems.** Reasoning is a process of arriving at a conclusion based on a collection of premises. Expert system based reasoning is one of the popular rule-based inferencing system that is in use today. This consists of three parts: a knowledge base (a set of rules and known facts); acquired data (derived facts and data); and an inference engine (reasoning logic).
- **Fuzzy Models:** Given the choice of system input and output variables, their linguistic modifiers with the associated fuzzy membership functions, an appropriate implication function, aggregation function, and defuzzification operator, if so desired, a fuzzy model that represents the system can be specified by a set of rules, their structure, and the fuzzy membership function parameters. Fuzzy systems model qualitative and quantitative non-linearity of systems. Attractive features include reduced design complexity, rapid prototyping, flexibility, simplicity, cost effectiveness, and inherent parallelism. This

explains the popularity of fuzzy systems in diverse application areas such as control, prediction, evaluation, cognition, analysis, and information management.

- **Neural Models:** Artificial Neural Networks (ANNs) are brain-inspired connectionist models that consist of many similar linear and nonlinear computational elements connected in complex patterns. The simple computational elements, also known as neurons, when associated in complex patterns, have the ability to perform tasks such as memory recall, pattern recognition, and learning. The ability of neural networks to learn from repeated exposure to system characteristics has made them a popular choice for many applications in image processing, system identification and control, pattern recognition and classification, financial prediction, and signal processing. Neural models can be sigmoidal networks, associative neural networks, radial basis function neural networks, or clustering networks. The choices are many.
- **Tree structures:** A tree structure is an algorithm for placing and locating objects in a database. The algorithm finds data by repeatedly making choices at decision points called nodes. A node can have branches (also called children) whose number can vary from two to several dozens. The structure is straightforward, but in terms of the number of nodes and children, a tree can be gigantic.

In aerospace applications, modeling is ubiquitous. Intelligent modeling tools are used for operator behavior modeling, flight test data modeling, aerodynamic modeling for design, etc.

3.1.1 Improved Robustness using Fuzzification

One of the benefits of fuzzy logic modeling is the fuzzy granularization that is obtained by defining fuzzy sets for the inputs and outputs of a system. Krishnakumar and Kulkarni (1998) outlined an approach that combines this granularization advantage with the knowledge available through an existing linear dynamic controller, to arrive at a powerful hybrid technique. This technique provides an outer fuzzy shell to existing control techniques and therefore combines the strengths of conventional as well as fuzzy control methodologies. The equivalent fuzzy dynamic controller is shown in Figure 1. It was also shown via a non-linear robustness analysis, that fuzzification of the inputs and outputs lead to better robustness to system uncertainties via the manipulation of the distance between the membership functions.

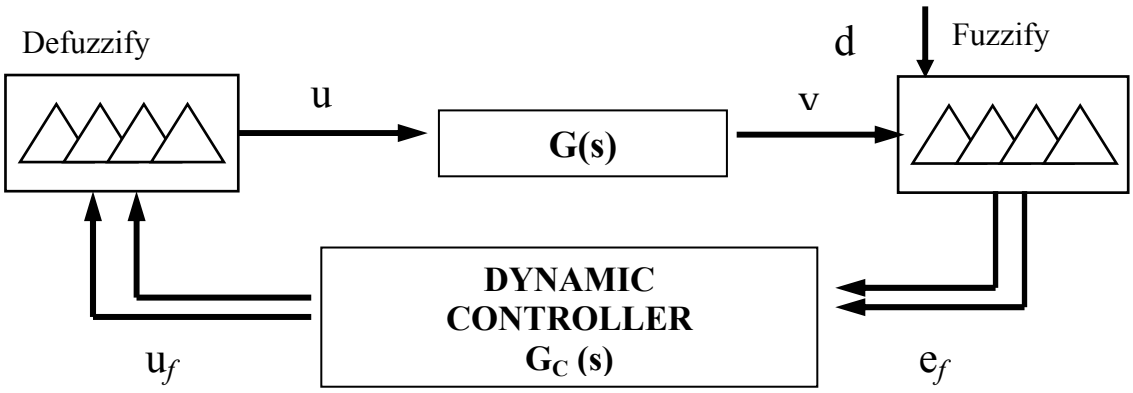


Figure 1. Equivalent Fuzzy Dynamic Controller Architecture

3.2 Intelligent Systems for Search

If the solution to a task can be represented by a set of N parameters, then the job of finding this solution can be thought of as a search in an N-dimensional space. This is referred to simply as the *search space*. More generally, if the solution to a task can be represented using a representation scheme, R, then the search space is the set of all possible configurations that may be represented in R.

Search techniques are also employed in problems where we want to determine if we can reach the desired goal state from an initial state, to minimize the cost in reaching the goal state, etc. The state-space approach is one of the primary representations for such problems. The state-space consists of an *initial state* and a set of *operators*. Application of the operators produces a sequence of new states called the *path.* A *goal test* is defined and tested to determine when a new state is the desired goal state.

Search techniques are routinely employed in combinatorial optimization. Some tasks involve combining a set of entities in a specific way (e.g. the task of building a combat tactics plan). A general combinatorial task involves deciding (a) the specifications of those entities (e.g. what type of aircraft, the opponents aircraft descriptions, number of aircraft, etc), and (b) the way in which those entities are brought together (e.g. various elementary tactics formations and their relative positions). If the resulting combination of entities can in some way be given a fitness score, then combinatorial optimization is the task of designing a set of entities, and deciding how they must be configured, so as to give maximum fitness. Some of the popular search techniques used by the intelligent systems community includes:

- **Golden Section**: The most used one-dimensional search technique that guarantees an optimum in a finite amount of time is the Golden section search technique. This technique belongs to the interval reduction family in which the interval is reduced by throwing out regions that definitely do not contain the optimum.

- **Breadth-first Search:** Search a state space by constructing a tree consisting of a set of leaves and branches. The algorithm defines a way to move through the tree structure.

- **Depth-first Search:** Instead of completely searching each level of the tree before going deeper, the algorithm follows a single branch of the tree down as many levels as possible until a solution or a dead-end is reached.

- **Heuristic Dynamic Programming (HDP):** HDP is based on an attempt to approximate Howard's form of the Bellman equation (Howard, 1960):

$$J(x_t) = \underset{u_t}{Min}\left\{ U_{PM}(x_t) + \gamma J(x_{t+1}) \middle|\ x_{t+1} = f(x_t, u_t, noise) \right\}$$

where x_t is the state vector, u_t is the control vector, $U_{PM}(.)$ is the one stage Performance Measure function, $f(.,.,.)$ is the model of the system, and $\gamma\,(0 < \gamma \le 1)$ is the discount factor. Variations of HDP are used in solving combinatorial optimization problems.

- **A*Search:** This is the most famous search algorithm used by the AI community. Here we combine the cost estimate *J(n)* of traversing from step *n* to the goal state and *U(n)* which is the known path cost from the start node to step *n*.

- **Genetic Algorithms:** A type of evolutionary computation devised by John Holland (Holland, 1975). A model of machine learning that uses a genetic/evolutionary metaphor. Implementations typically use fixed-length character strings to represent their genetic information, together with a population of individuals which undergo crossover and mutation in order to find interesting regions of the search space.

In aerospace applications, intelligent search techniques are used typically for searching a design space, searching for an optimal scheduling and planning schemes, and for solving combinatorial optimization problems. In the example presented next, we examine the use of genetic algorithms for finding optimal tactics formation.

3.2.1 Application: Search for Optimal Air Combat Tactics

The complete design and specification of air combat tactics for many-vs.-many engagements poses a considerable challenge to tactical planners. While solutions have been developed for one-vs.-one or few-vs.-few encounters, the results may not generalize to larger engagements where formation tactics become increasingly important. The most effective formation tactics employ a basic fighting unit of two aircraft (called a *section* or *element*) (Shaw, 1988). Since this is how all fighter pilots learn their craft, it would be most effective for optimized tactics not to deviate from this established tactical doctrine. Accordingly, software tactics modules that employ basic fighting units of two aircraft were developed. Since large numbers of fighter aircraft are difficult to control, formation tactics for large groups may be developed using a hierarchical structure consisting of smaller units or divisions; for example, a four-airplane division called the *fluid four* consists of two elements. Each element consists of two aircraft, but they are treated as a unit. This hierarchical concept is used to develop a GA-based approach to search for optimal air combat tactics. Given a palette of air combat maneuvers and standard small-formation tactics as building blocks, GAs are used to determine how they can be integrated to produce large fighting groups that optimize overall combat effectiveness.

Tactics implementation proceeds as follows (The complete simulation environment is presented in Figure 2. For more details, see (Mulgund et al, 2001)):

- Define a set of commonly-used element and division formations as well as the underlying tactical maneuvers and attack tactics.
- Develop a set of principles for aggregating the small formation tactics for large MvN engagements, and implement a method for doing so in the GA software. To illustrate, consider a team consisting of four aircraft. Using only the fighting wing and double-attack, the possible team formations are shown in Figure 3 (assuming both elements use the same two-ship formation). A similar approach can be used to develop large division formations from smaller 2-ship and 4-ship groupings.
- Use the resultant formation tactics to drive the engagement, and evaluate the results via the performance metric generator.

- Search for the best MvN engagement tactics so as to optimize the performance metrics.

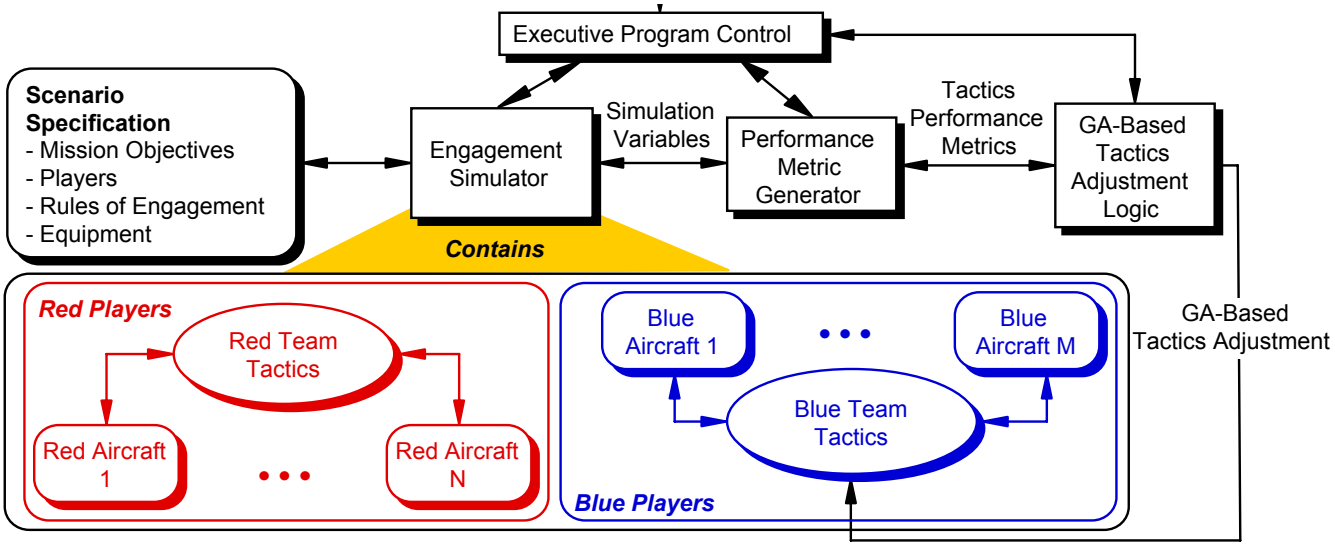


Figure 2. Combat Tactics Design Environment

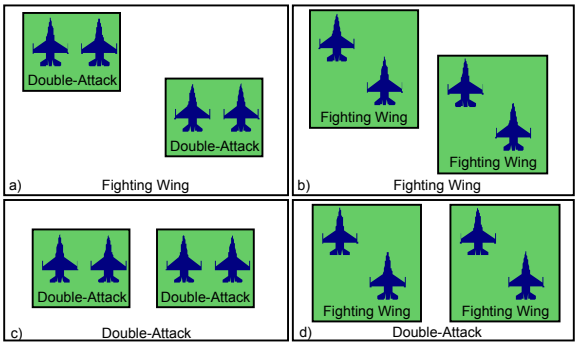


Figure 3. Potential Four-Aircraft Formations.

3.3 Intelligent Systems for Design

Aerospace systems are complex in nature and their design interdisciplinary. For example, design optimization conducted individually on subsystems such as wing, propulsion, and automatic control will not integrate without extensive redesign. Even individual design of these subsystems require interaction of several subsystems. Intelligent systems can help in many ways to enhance this experience. Several researchers have shown the benefits of neural networks and fuzzy logic for representation of known data and designer expertise. Genetic algorithms have been routinely used for searching through large spaces with several constraints and subsystem interactions. A generic representation of the design philosophy using intelligent system features are shown in Figure 4. In the design application presented next, Neural networks are used for representation of data (modeling) and genetic algorithms are used to search for the type of inputs to be used for the neural networks.

3.3.1 Application: Engine Estimator Design

One of the common objectives of aircraft engine control is to enhance engine performance under deteriorated conditions. To maximize engine performance efficiently under degraded conditions, a fault tolerant engine control scheme can be applied. The first step to implement the fault tolerant engine control architecture is developing an engine performance estimator. This application focuses on designing an engine performance estimator using a combination of a genetic algorithm (GA) and a radial basis function neural network (RBFNN) for the implementation.

Generally, traditional engine performance estimators, such Kalman filter estimator, involves intensive computational procedures because of engines' physical complexity which requires a large number of measurements to be taken and processed. To overcome computational complexity, model estimation using neural networks has emerged. Neural network-based model estimation has been applied to areas such as optics, robotics, and system control. Attracted by the advantages of neural networks, the recent studies of fault tolerance have employed neural network architectures. In these studies, the input selection is executed by simple inspection of data files. This manual inspection can be replaced by automatic inspection using GAs. The need for the design that involves selecting the best inputs is driven by the cost of sensors. In this case fewer the sensors the better the cost.

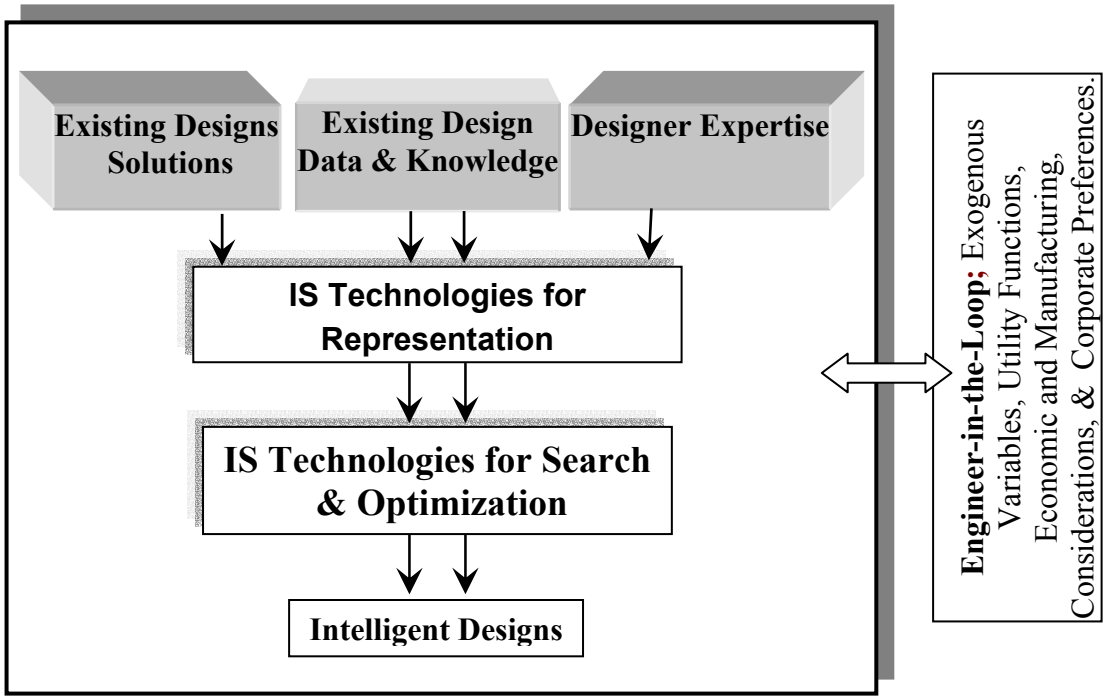


Figure 4. An Intelligent Design Approach

To apply GAs to this problem, the parameter performance can be evaluated by an objective function (performance index or PI). Regularly, a quadratic error measure is used for the objective function. The error is defined as the difference between desired performance and the current design's performance.

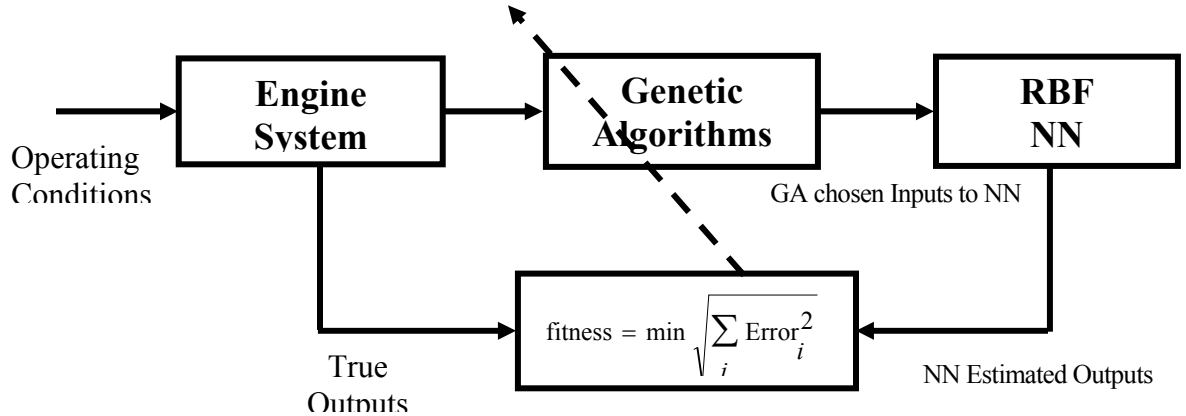


Figure 4. GA-RBFNN Architecture

GA-RBFNN Architecture: The GA-RBFNN architecture is shown in Figure 5. The Genetic Algorithms picks "n" inputs out of the possible 21 variables shown in table 1. The quantity "n" is a user choice. For example, if n=5, the GA has the following chromosome representation

10110	10011	11011	10111	00110
Input 1	Input 2	Input 3	Input 4	Input 5

We use a 5 bit representation to cover the 21 possibilities. Since 5 bits give us 32 possibilities, there will be multiple mapping for some of the variables.

The data from the first 150 seconds (300 data points with 0.5 seconds of step size) is used for the RBFNN's training. For the GA fitness, all the 500 data points are utilized. This way, we incorporate a combination of training and validation data into the GA fitness.

The RBFNN produces one output, either compressor **stall margin (*sm27*)** or **thrust (*fn*)**. The output is then compared with a desired performance of the corresponding performance measure (desired *sm27* or *fn*). The difference between the output from the RBFNN and the desired performance becomes the estimation error. Squaring the error and summing it up over the time range (500 data points) results in a fitness function of the entire system. For more details, please see Krishnakumar and Hachisako (2000).

3.4 Intelligent Systems for Control

There are two main aspects to intelligent control: (1) the "intelligence" to analyze the changing environment; and (2) the resources to respond to the changing environment. Intelligence connotes the analytical ability to comprehend and react to the changing environment. Resources connote the physical components of the system that are necessary to react to the environment. In this work, we concentrate on the need to harvest and interpret the information from the network of sensors and to apply it for control such that good performance is maintained under any of the following situations:

- Loss of control due to failure
- Aircraft characteristics change due to damage (center of gravity, inertia, etc.)

- Changing operating conditions (altitude, mach, etc.)
- Environmental effects due to wind and turbulence

Intelligent control applications focus on control problems that otherwise cannot be solved, or cannot be solved in a satisfactory way by traditional control techniques alone. Intelligent control as practiced today encompasses many fields from conventional control such as optimal control, robust control, stochastic control, linear control, and nonlinear control, as well as the more recent fuzzy, genetic, and neuro-control technologies. In the next subsection, intelligent control is classified based on the idea of self-improvement as the goal toward higher levels of intelligence.

3.4.1 Application: Levels of Intelligent Control

In a general sense, an intelligent controller design can be stated as the following:

given the dynamic system as:

$X(t+1) = f(X(t),U(t),t)+\eta$; where X are the state variables, U is the control vector, and η is an unknown disturbance

a set of goals generated as a function of time as

$Xg(t+1) = g(Xg(t),X(t),t)$

a performance measure as:

$J(t+1)= \Im(M(Xg(t),X(t),U(t),t))$; where $\Im$ is an operator (usually summation over T)

and a planning function as

$P(t+1) = p(X(t),P(t),t,\nu)$; where ν is system faults and emergencies

the intelligent controller needs to arrive at a control, U(t), such that the system (in the order of priority)

- is locally stable (includes handling quality of the system)
- follows closely the desired path (closeness defined by the performance measure)
- constantly optimizes long-term and short-term goals
- reacts to changing environments by properly adapting the planning functionality.

Over the past decade, several innovative control architectures utilizing the intelligent control tools have been proposed. We believe that a practical way to accommodate the above needs is to approach the system as having various levels of capabilities for self-improvement. Self-improvement is an important goal of human intelligence. Self-improvement is quantifiable and measurable in various ways. By defining intelligent Control with various levels of intelligence, the definition is left 'open ended' such that it will not become obsolete, and it will accommodate easily the innovations that will inevitably come from the contributions of such fields as cognitive science, computer hardware, sensors and actuators, learning theory, and control architectures.

KrishnaKumar (1997) has proposed a classification scheme based on the ability of the control architecture for self-improvement (see Table 1). The classification scheme divides the control architectures among levels of intelligent control (LIC). For instance, most of the proposed architectures can be divided among level 0, level 1, level 2, and level 3 intelligent control schemes. Based on this classification scheme, several seemingly differing control architectures can be looked at as achieving similar goals.

Table 1. The Levels of Intelligent Control

Level	Self improvement of	Description
0	Tracking Error (TE)	Robust Feedback Control: Error tends to zero.
1	TE + Control Parameters (CP)	Adaptive Control: Robust feedback control with adaptive control parameters (error tends to zero for non-nominal operations; feedback control is self improving).
2	TE+CP+ Performance Measure (PM)	Optimal Control: Robust, adaptive feedback control that minimizes or maximizes a utility function over time.
3	TE+CP+PM+ Planning Function	Planning Control: Level 2 + the ability to plan ahead of time for uncertain situations, simulate, and model uncertainties.

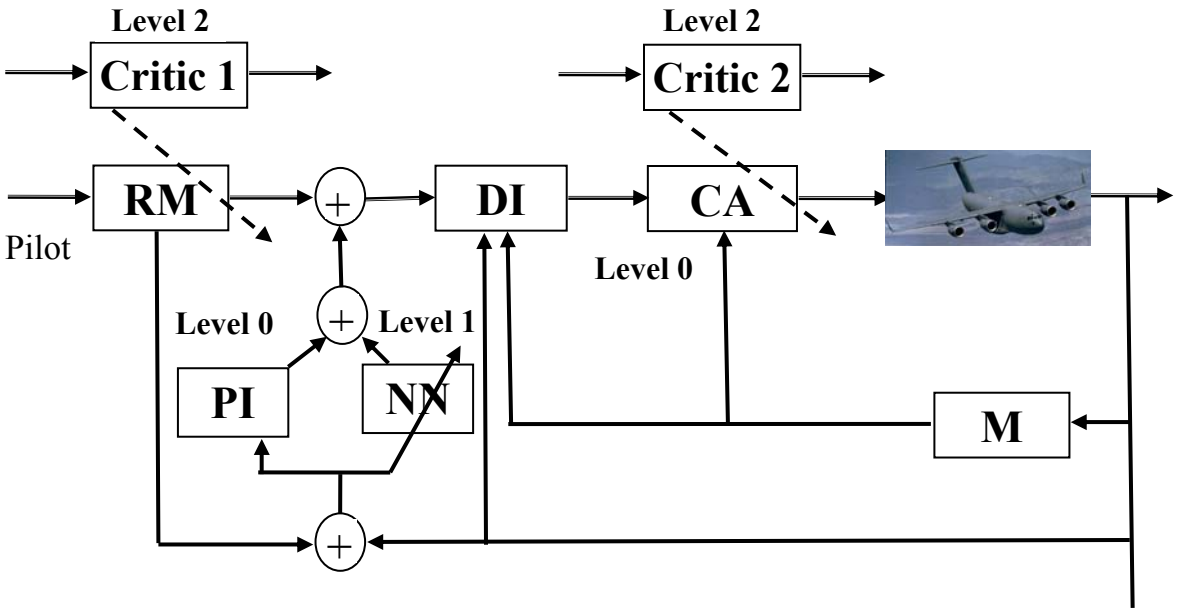
Level 0 Intelligent Control -- A Robust Controller: Self-improvement of Tracking Error (TE) is an important goal of many control techniques. To achieve this, one designs robust feedback controllers with constant gains that improve the error as time goes to infinity. We consider this as "Level 0 Intelligent Control".

Level 1 Intelligent Control -- An Adaptive Controller: Self-improvement of control parameters towards the goal of achieving better tracking error or some error oriented goal, is the next level in intelligent control. We consider this as "Level 1 intelligent Control". This level is essentially a robust feedback controller with adaptive parameters that helps the error tend to zero for non-nominal operations and feedback controller is self improving.

Level 2 Intelligent Control -- An Optimal Controller: Self-improvement of an estimate of the performance error (or some measure of performance over time) towards the goal of minimization or maximization of an utility function over time (error tends to zero and a measure of performance is optimized) is the next level in intelligent control. We consider this as "Level 2 Intelligent Control". This level is essentially a robust feedback controller with adaptive parameters that helps the error tend to zero for non-nominal operations, feedback controller is self improving, and a measure of performance that is self-improving is optimized over time.

Level 3 Intelligent Control -- A Planning Controller: In addition to level 2 capabilities, Level 3 Intelligent Control includes self- improvement of planning functions. Planning functions include contingency planning, planning for emergencies, planning for faults, etc. These planning functions could be static for Level 2 but needs to be self-improving for Level 3.

Figure 6 presents the implementation of a Level 2 Intelligent Control on a C-17 test bed at NASA Ames research center. The levels as outlined earlier are labeled in the figure. It should be noted that Level 0 is non-adaptive whereas Level 1 is adaptive. Level 1 is non-optimal whereas Level 2 is optimal.



RM: Reference Models: The pilot commands roll rate and aerodynamic normal and lateral accelerations through stick and rudder pedal inputs. These commands are then transformed into body-axis rate commands, which also include turn coordination, level turn compensation, and yaw-dampening terms. First-order reference models are used to filter these commands in order to shape desired handing qualities.

PI Error Controller: Errors in roll rate, pitch rate, and yaw rate responses can be caused by inaccuracies in aerodynamic estimates and model inversion. Unidentified damage or failures can also introduce additional errors. In order to achieve a rate-command-attitude-hold (RCAH) system, a proportional-integral (PI) error controller is used to correct for errors detected from roll rate, pitch rate, and yaw rate (p, q, r) feedback.

NN: On-Line Learning Neural Networks: The on-line learning neural networks work in conjunction with the error controller. By recognizing patterns in the behavior of the error, the neural networks can learn to remove biases through control augmentation commands. These commands prevent the integrators from having to windup to remove error biases. By allowing integrators to operate at nominal levels, the neural networks enable the controller to provide consistent handling qualities.

DI: Dynamic Inversion: Dynamic inversion is based upon feedback linearization theory. No gain-scheduling is required, since gains are functions of aerodynamic stability and control derivative estimates and sensor feedback. To perform the model inversion, acceleration commands are used to replace the actual accelerations in the quasi-linear model. The model is then inverted to solve for the necessary control surface commands

CA: Control Allocation: An optimal control allocation technique is used to ensure that conventional flight control surfaces will be utilized under normal operating conditions. Unconventional flight control surface allocations are only utilized when the primary flight control surface commands exceed the known limits of deflection. For example, in the longitudinal axis pitch rate control is normally provided through symmetric elevator deflections. If this command should saturate, then the remaining portion of the command is applied to symmetric ailerons. If the symmetric aileron command saturates, then the remaining portion of that command is applied to symmetric thrust. The symmetric aileron command is limited, by the differential aileron command, so that secondary pitch control does not interfere with primary roll control.

M: Pre-Trained Neural Network Model: A Levenberg-Marquardt (LM) multi-layer perceptron is used to provide dynamic estimates for model inversion. The LM network is pre-trained with stability and control derivative data generated by a Rapid Aircraft Modeler. This block can be replaced by other on-line derivative (parameter) estimation techniques.

Critic 1 and 2: Adaptive Critics are utilized to optimize the control allocation scheme and to shape the reference model dynamics in the event of a failure.

Figure 6. Level 2 Intelligent Flight Control System

3.5 Intelligent Systems for Training

Many characteristics of intelligent systems are readily applicable to training. Although training entails many different areas of aerospace engineering, our experience has been with pilot training. Figure 7 presents an implementation of an automated hover training system. This system was implemented in a fixed-base simulation facility and was shown to provide basic hover training skills with no human intervention. The neural network based intelligent system adapts the helicopter dynamics to the student pilot and automatically changes the dynamics of the helicopter as learning progresses. For more details, please see KrishnaKumar et al (1994)

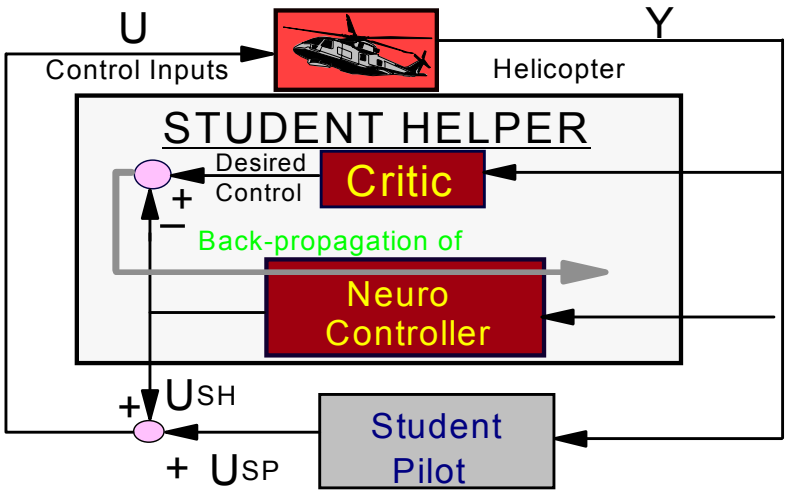


Figure 7. Automated Training Using an Intelligent System

3.6 Intelligent Systems for Autonomous Agents

An Agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors. An autonomous agent is shown in Figure 8. A rational agent does the right thing. How successful the agent is measured using a performance measure that is defined using goals. It is assumed that the agent has some percepts to satisfy observability or sensing requirements. Knowledge of the environment (Model) and actions that can be performed (Controllability) are implicit to the agent. Some of these are pre-programmed and some are learnt on-line. An ideal rational agent optimizes the measure of performance. Intelligent systems are used in different capacities such as knowledge representation, adaptability, agent-to-agent interaction, planning, and optimization. For more details on agents, see reference (Russell and Norvig, 1995).

3.7 Other Areas of Application

Intelligent systems can be embedded in almost any application where information needs to be processed to provide a usable output. Some areas that were not specifically addressed earlier include:

- Intelligent Systems for Decision Making
- Intelligent systems for planning and scheduling
- Intelligent systems for health management
- Intelligent systems for prediction
- Intelligent systems for knowledge discovery

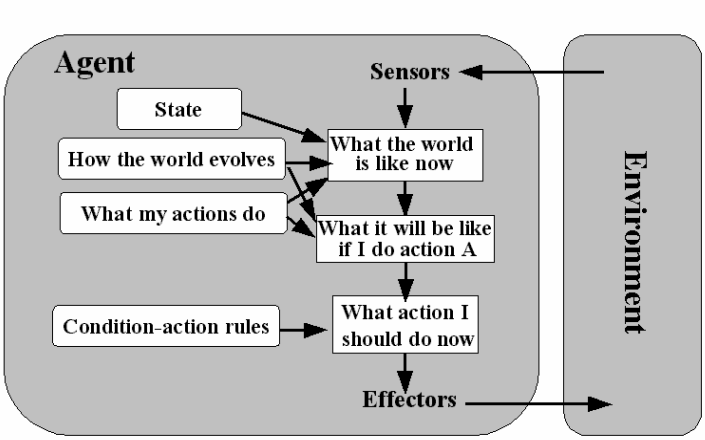


Figure 8. Intelligent Autonomous Agent Architecture (Russell and Norvig, 1995).

4 Future of Intelligent System Applications

Intelligent systems provide a means by which complex problems can be addressed and in many cases solved to a satisfactory level. The benefits can be categorized as either immediate or in-the-future. The immediate benefits are in applications of intelligent systems to areas where existing methodologies are marginally satisfactory and incorporating intelligent systems provide better efficiencies and solutions. Examples include: inverse design, adaptive control, optimal search, etc. The future benefits are more exciting. Intelligent systems will help formulate and solve problems such as brain-like control and decision-making, human-machine collaborative work, instant speech recognition, thought control, human capability enhancement, advanced pattern recognition, real-time scheduling, automated design, intelligent maneuvering for unmanned aerial vehicles, and autonomous security search. On a cautious note, intelligent system researchers should examine closely the analytical framework of their innovations. Analytical framework along with standardization has been shown to be important for the ultimate ticket to real implementations in aircraft applications.

5 Acknowledgement

Part of the work reported here was conducted while the author was a faculty at the University of Alabama. All of the students and colleagues who worked on these projects are hereby acknowledged for their contributions to the work highlighted above.

6 References

Holland, J. H. (1975). *Adaption in Natural and Artificial Systems*, University of Michigan Press.

Howard, R. A. (1960). *Dynamic Programming and Markov Process*. MIT Press, Cambridge, Massachusetts.

KrishnaKumar, K. & Hachisako, Y. (2000). *Two Applications of Genetic Algorithms, GA for optimization in Aeronautics and Turbomachinery*, KrishnaKumar, K. (1997). *Levels of Intelligent Control*, AIAA Tutorial at New Orleans, LA, August.

Krishnakumar, K. (2000). *Intelligent Control for Aerospace Systems*, Global Aerospace Technology, World Market Research Center.

Krishnakumar, K.S., Kulkarni, N., "Equivalent Dynamic Fuzzy Controllers and their Application to Engine Control", 34th AIAA Joint Propulsion Conference, Cleveland, Ohio, July, 1998.

KrishnaKumar, K., Sawhney, S., and Wai, R. (1994). *Neuro-controllers for adaptive helicopter hover training*, IEEE Transactions on Systems, Man, and Cybernetics, August, 1994 Vol. 24, No. 8. pp. 1142-1152.

Kurzweil, R. (1990). *The Age of Intelligent Machines*. MIT Press, Cambridge, Massachusetts.

Luger, G. F. and Stubblefield, W. A. (1992). *Artificial Intelligence*, Benjamin-Cummings, Redwood City, California

Mulgund, S., Harper, K., Krishnakumar, K., & Zacharias, G. (2001). *Air Combat Tactics Optimization Using Genetic Algorithms*, AIAA Journal of Guidance Control and Dynamics,

Rich, E. and Knight, K. (1991). *Artificial Intelligence,* McGraw Hill, New York.

Russell, S. and Norvig, P. (1995). *Artificial Intelligence: A Modern Approach*, Prentice Hall, New Jersey.

Schalkoff, R. J. (1990). *Artificial Intelligence: An Engineering Approach*, McGraw Hill, New York.

Shaw, R. (1988). *Fighter Combat: Tactics and Maneuvering*, Annapolis, MD: Naval Institute Press.